COMP9517

Lab 1, T1, 2020

This lab presents a revision of important concepts from week 1 and 2 lectures. Most questions require you to use OpenCV, an open source software package that is widely used in this field.

The last question (Question 5) is assessable AFTER THE LAB. Please submit the results (including the high/low-pass filtered images) in a zip file via webCMS3 **by 11:59:59 AM (close to noon) on Wednesday March 4th, 2020**.

Sample images used for these questions are available in the link 'Sample Images'.

QUESTIONS SNAPSHOT

Use this snapshot to plan your time. The detailed questions, along with a revision of concepts and hints to solve the questions, appear later.

1. Read a grey scale image and perform contrast stretching to improve the visual quality of the image.
2. Write a function that computes the histogram of a grey scale image and displays a plot. Then, for a given image, match its histogram to the histogram of the given reference image.
3. With the given image, use the Sobel operator to compute the image gradients at X and Y directions.
4. Implement a mean filter, median filter and a gaussian filter. Perform noise removal on the images provided in the sample images.
5. Read two images, perform high pass filtering on one and low pass filtering on the other. Create a hybrid image by adding the two results. The answer to this question will be assessed.

Contrast Stretching

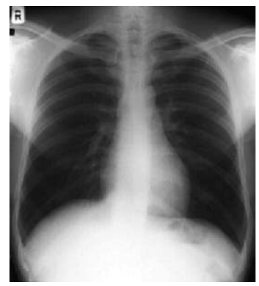
Contrast in an image is a measure of the range of intensity values within an image, and is the difference between the maximum and minimum pixel values. The full contrast of an 8-bit image is 255(max)-0(min)=255, and anything less than that results in a lower contrast image. Contrast stretching attempts to improve the contrast of an image by stretching (linear scaling) the range of intensity values.

Assume that ***Or*** is the original image and ***Tr*** is the transformed image. Let **a** and **b** are the min and max pixel values allowed in an image (8-bit image, a=0 and b=255), and let **c** and **d** be the min and max pixel values in a given image, then the contrast stretched image is given by the function:

$$Tr = (Or - c)\left(\frac{b-a}{d-c}\right) + a$$

QUESTION 1: Read a grey scale image (imageQ1.jpg) and perform contrast stretching to improve the quality of the image. Example shown below is a poor contrast X ray image and its contrast stretched version.

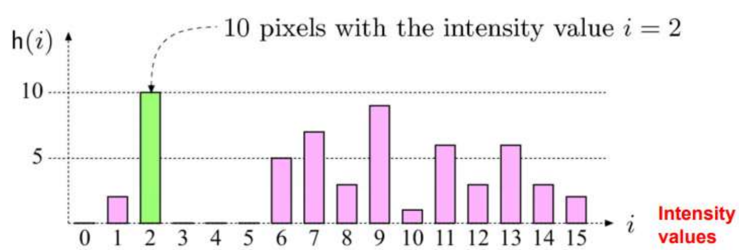
Original Image



Contrast Stretched image

Histogram

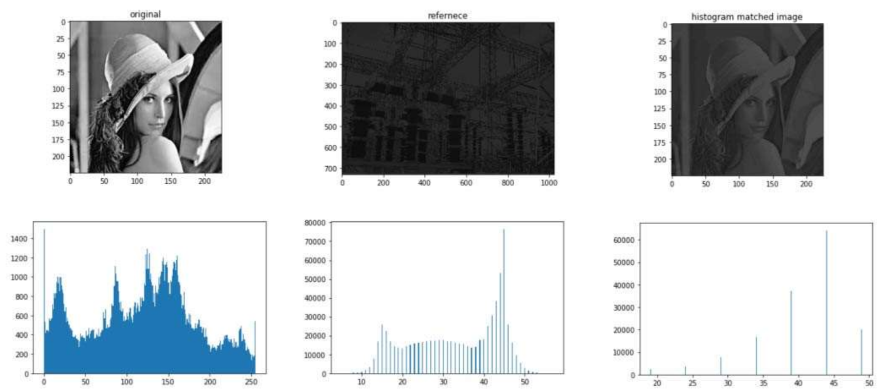
Histogram of an image shows the frequency of pixel intensity values. It only gives statistical information and nothing about the location of the pixels. For a digital image with grey levels from 0 to L-1, the histogram is a discrete function $h(Or_k) = n_k$, where Or_k is the kth grey level and n_k is the number of pixels with a grey level r_k.



h(i)	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 1: Histogram (Picture from [3]).

Histogram matching is the transformation of an image so that its histogram matches a specified histogram. Example shown below is an original image and its histogram matched version to a reference image.



QUESTION 2.1 Write a function that computes the histogram of a given grey scale image (imageQ21.jpg) and displays a plot. Better not use in built OpenCV functions.

QUESTION 2.2 For a given image (imageQ22.jpg), match its histogram to the one of the given reference image (imageQ21.jpg).

IMAGE FILTERING

Image filtering is a neighbourhood operation for modifying or enhancing an image, that can emphasise certain features or remove certain unwanted features. Image filtering is usually implemented as a convolution. Convolution is a mathematical operation which takes as input two functions (say ***f*** and ***g***) and produces an output function (say ***h***) which typically is a modified version of one of the original functions. In our case, ***f*** is the input image, ***g*** is the convolution mask/kernel, ***h*** is the filtered version of ***f*** and convolution creates weighted sums of the image pixels.

Image Edges

Edges are an important source of semantic information in images, and they occur in human visual perception at divisions between different areas of brightness, colour and texture. A grey scale image can be thought of as a 2D representation of heights and areas of different brightness lives at different heights. A transition between different areas of brightness in an image ***I***, means there must be a steep slope which we formalise as the gradient of

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$$

of the Image. Now our image ***I*** is discrete so we approximate the continuous quantities $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ by finite difference kernels. A simple example of a finite difference kernel is the Sobel filter (F_x and F_y), which is the subject of the following question.

QUESTION 3: With the given image (imgeQ3.jpg), use the Sobel operator to compute the image gradients at x and y directions. To do this, first define the 2D filters (F_x and F_y). Then perform convolution between the image and F_x to obtain the gradients at x direction, and similarly perform convolution between the image and F_y to obtain the gradients at y direction.

$$F\_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

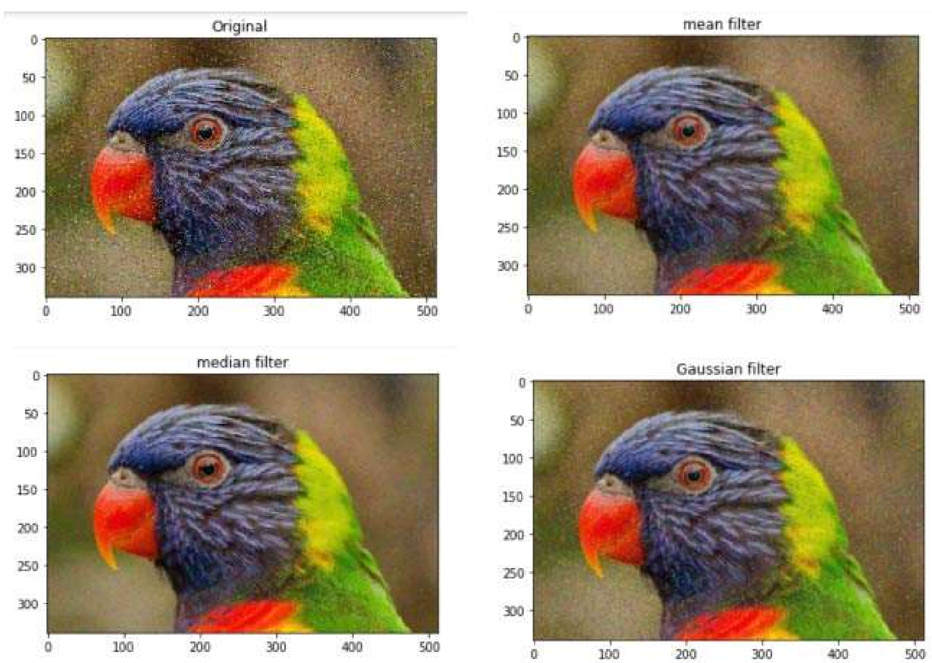
$$F\_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Notes:

- The OpenCV built-in Sobel functions can be applied to achieve the result. This can be a way of verifying the gradient outputs.

Smoothing Filters (Low-pass Filters)

Noise (random variations in intensity) removal from images is implemented using Image filtering. Salt and pepper noise, impulse noise and gaussian noise are some of the commonly observed types of noise. Different types of filters are suited for different types of noise. A smoothing filter is often called a low-pass filter as it allows low frequency components of the image to pass though (regions with similar intensity values) but stops high frequency components from passing through (edges or noise). Shown below are the results of applying different filters on an image corrupted by salt and pepper noise. It can be observed that the median filter produces the best result in this case.



QUESTION 4: Implement a mean filter, median filter and a gaussian filter. Perform noise removal on the given image (imageQ4.jpg). Try with different filters and kernel sizes, observe the variations and find the best filter and kernel size for the noisy images.

Image Sharpening using High-pass filters

Image sharpening is used to highlight fine details in an image or to enhance features that are blurred. High pass filters allow high frequency components such as edges and noise to pass through, while blocking low frequency components. Shown below are the results of applying a high-pass filter and the resulting sharpened image.

Hybrid Images

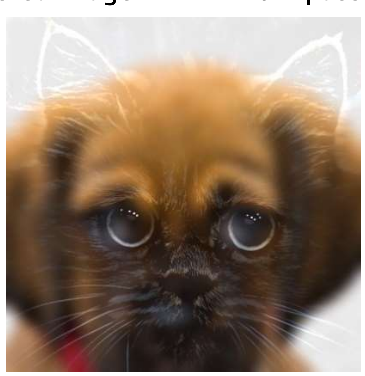
High pass filters and low pass filters are the two main types of image filters. Images filtered with a low pass filter are smoothened, while those filtered with high pass filters are sharpened. If a high-pass filtered image is combined with a low pass filtered image, a hybrid image is formed which changes its content as the distance between the image and the viewer changes. Shown below are two sample images, their high-pass filtered, and low-pass filtered versions and the resulting hybrid image.

Original Image 1

Original Image 2

High-pass filtered Image

Low-pass filtered Image



Hybrid Image

Assessment Question

QUESTION 5: Read two images (imgeQ51.jpg and imageQ52.jpg), perform high pass filtering on one and low pass filtering on the other. Create a hybrid image by adding the two results.

REFERENCES

[1]. Krig S. (2014) Image Pre-Processing. In: Computer Vision Metrics. Apress, Berkeley, CA, https://link.springer.com/chapter/10.1007/978-1-4302-5930-5_2#citeas

[2]. Aude Oliva, Antonio Torralba and Philippe G. Schyns (2006). "Hybrid images" (PDF). ACM Transactions on Graphics (SIGGRAPH 2006 issue). 25 (3): 527–532.